George Wahsington University

Comparison of Local Explanation Techniques for Convolutional Neural Network Models

Ya Liu
Advisor: Amir Jafari

December 2020

List of Contents

1 Introduction

Users need to understand why a model makes a decision and decide whether to trust its prediction. Some models are intrinsically interpretable but not very powerful, e.g., linear models, whereas deep neural network models are powerful but too complex to interpret. Deep neural network models usually have a high accuracy. However, when it comes to application in practice, duo to complexity of these models, they are often blamed for lack of interpretability. Interpretability is the degree to which a human can understand the cause of a decision. Thus, there is a tension between accuracy and interpretability.

In response, numerous explanation techniques are designed to help users understand and interpret these models. These techniques can be classified into two categories, local explanation and global explanation. Local explanation aims to explain each prediction a model makes, while global explanation aims to explain the whole model. They can also be classified into model-specific methods, which can only be used for specific kinds of models, and model-agnostic methods, which can be used for multiple kinds of models. Figure 1 shows the classification of most explanation methods.

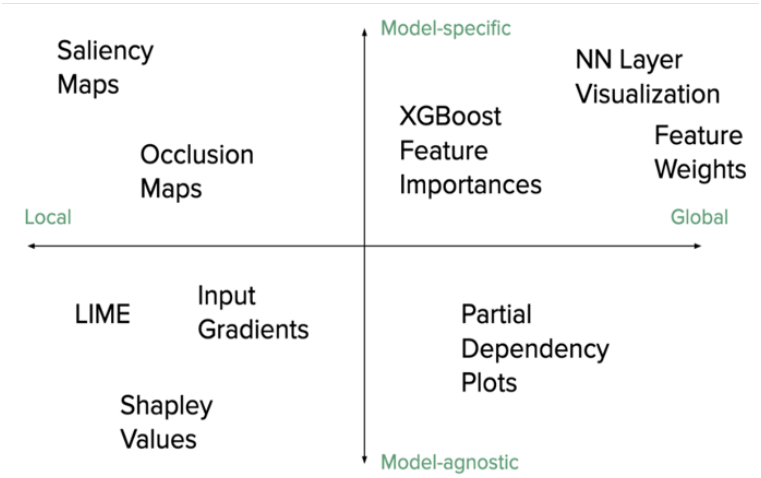

Figure 1: Classification of Explanation Techniques

Even though numerous methods are proposed to explain models, it still remains unclear that when one method is preferable over another. Since this is a broad topic, I try to narrow down this problem according to two criteria. First, I consider local explanation methods; Second, I focus on image classification problems and thus explain convolutional neural network (CNN) models only. Accordingly, three techniques are selected, DeepLIFT, SHAP, LIME.

In the following sections, this report briefly introduces these three techniques, builds two CNN models, and compares explanations of three techniques based on these models and same inputs.

The GitHub link of this project can be found here.

2 Three Explanation Techniques

In this section, I briefly discuss the core ideas behind these three explanation techniques.

2.1 DeepLIFT

DeepLIFT (Deep Learning Important FeaTures) is a gradient based algorithm. This kind of algorithm uses gradient to represent feature importance because gradients can show how the model's prediction changes as the feature changes. The feature importance can be calculated by

$$\text{feature importance} = x_i \times \frac{\partial Y}{\partial x_i} \tag{1}$$

However, one limitation of gradient based algorithms is gradient saturation problem, which means the feature has taken a value which has the most impact on the score globally, but the gradient fails to capture this importance. For example, at the green point, the feature importance is zero, which obviously does not make sense since x is an important feature for y.

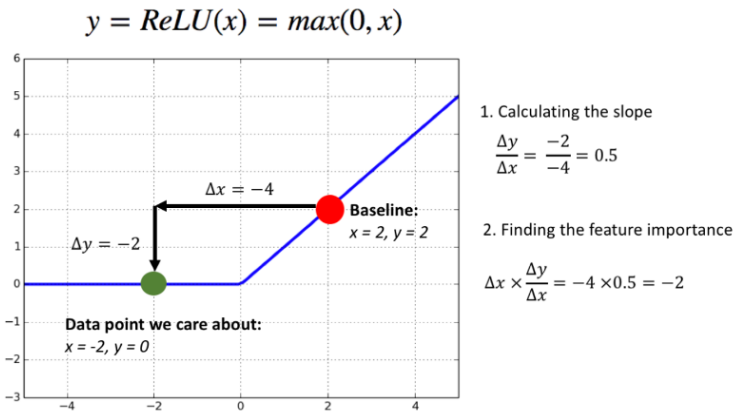

Figure 2: ReLu Function

In response, DeepLIFT is proposed to solve this problem. Instead of using gradient, it utilizes slope to represent feature importance. Gradient describes how y changes as x changes at the point x, while Slope describes how y changes as x differs from the baseline. Slope can be calculated by

$$\text{feature importance} = (x_i - x_i^{baseline}) \times \frac{Y - Y^{baseline}}{x_i - x^{baseline}} \tag{2}$$

Taking this ReLu function for example, we can see that the slope is 0.5, and feature importance is -2. In this way, DeepLIFT solves gradient saturation problem. It is also worth noticing that in practice, we usually use the mean of the data set or an array of zeros as baseline.

2.2 SHAP

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It argues that six explanation methods are a transformation of additive feature attribution methods. Unified with Shapley value, a concept derived from cooperative game theory and computed to measures each person's contribution among coalition, various explainers are proposed in terms of different kinds of models. For example, TreeExplainer explains the output of ensemble tree models; AdditiveExplainer computes SHAP values for generalized additive models. Among those explainers, three (KernelExplainer, DeepExplainer, GradientExplainer) are selected since this paper focuses on CNN models.

KernelExplainer has no assumption about models, and is a combination of Linear LIME and Shapley values. It uses a special weighted linear regression to compute the importance of each feature.

GradientExplainer is only used for deep neural network models, and is extension of integrated gradients. As an adaptation to make integrated gradients values approximate SHAP values, expected gradients reformulates the integral as an expectation and combines that expectation with sampling reference values from the background dataset. This leads to a single combined expectation of gradients that converges to attributions that sum to the difference between the expected model output and the current output.

DeepExplainer is also designed for deep neural network models and is an enhanced version of the DeepLIFT algorithm. It approximates the conditional expectations of SHAP values using a selection of background samples. The per node attribution rules in DeepLIFT can be chosen to approximate Shapley values. By integrating over many background samples, Deep estimates approximate SHAP values such that they sum up to the difference between the expected model output on the passed background samples and the current model output.

2.3 LIME

LIME (Local Interpretable Model-Agnostic Explanations) a technique to explain the predictions of any machine learning classifier. The core idea behind is approximating the underlying model by an interpretable one (e.g., linear model).

In terms of explanation of images, it first divides an image into interpretable components (contiguous superpixels) as it is shown in Figure 3. Following the process in Figure 4, then it generates a data set of perturbed instances by turning some of the interpretable components “off”. For each perturbed instance, we get the probability that an object is in the image according to the model. We then learn a simple (linear) model on this data set, which is locally weighted. Finally, we present the superpixels with highest positive weights as an explanation, graying out everything else.

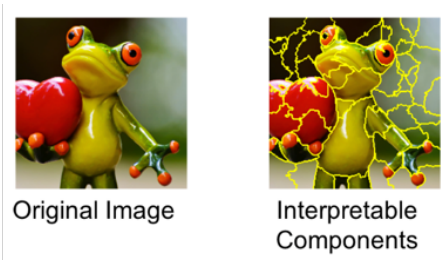

Figure 3: Superpixels

3 Two CNN Models

Two CNN models are built to be explained.

One is trained on handwritten digits image dataset from MNIST. The model structure is shown in Figure 5. Figure 6 is confusion matrix of predictions of test dataset, and it illustrates that the

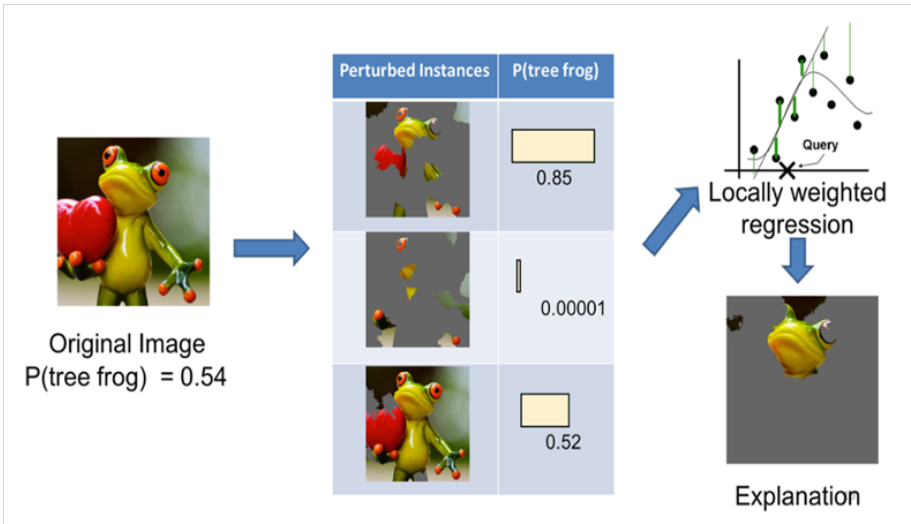

Figure 4: Core Idea of LIME

model has a good performance. Besides, I also finetune VGG16 on fashion product dataset. This

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])
```

Figure 5: Model Structure

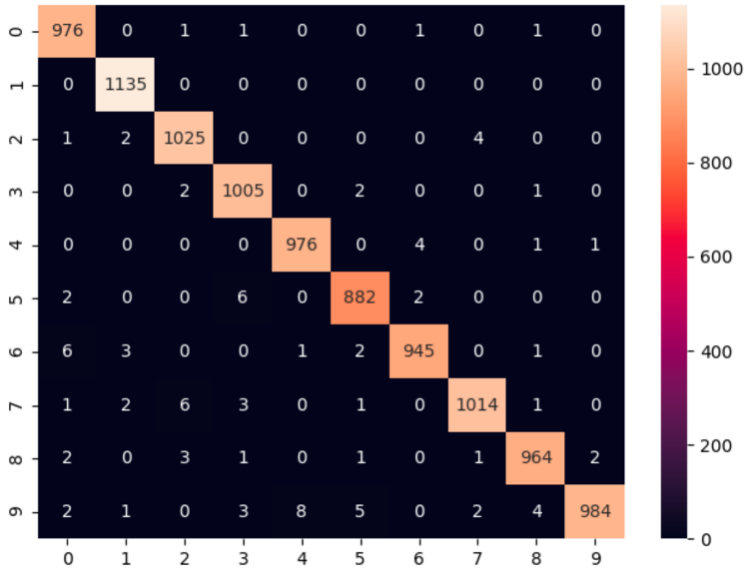

Figure 6: Confusion Matrix

dataset includes 143 classes, and I only choose five of them with each has about 1000 samples, and Figure 7 shows some images regarding these five classes. By simplifying this dataset, I try

to develop a model with high accuracy and meanwhile reduce the influence of errors on the performance of explanation techniques. We can aslo notice that casual shoes and flip flops are two similar classes, and thus it is interesting to see how models differentiate these two and how explanation methods explain them. Figure 8 is confusion matrix which indicates that this model has a high accuracy.

Figure 7: Confusion Matrix

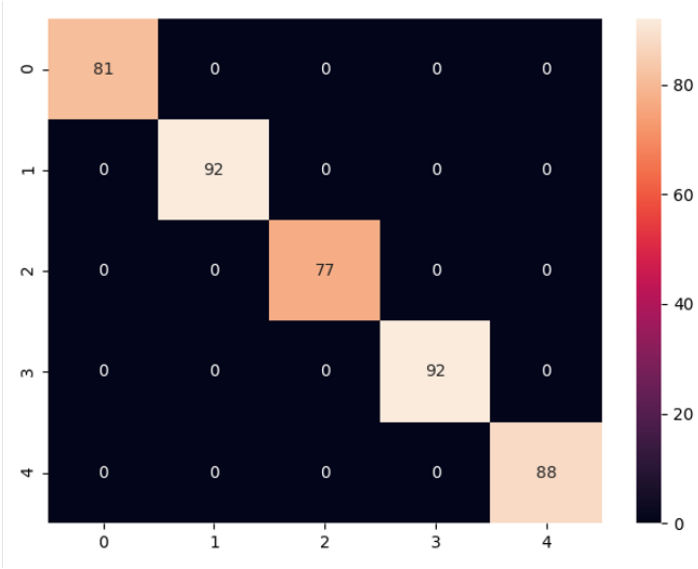

Figure 8: Confusion Matrix

4 Explanation Results

This section presents explanations of three methods on the two CNN models.

4.1 Experiment Setup

The whole project is run in Python, and all three explanation methods can be installed by using pip. It is worthwhile to notice that the version of Tensorflow and Keras are different (DeepLIFT: Tensorflow == 1.15.2, Keras == 2.3.1; SHAP: Tensorflow==2.3.0, Keras == 2.4.3; LIME: Keras == 2.4.3).

4.2 DeepLIFT

Figure 9 shows the explanation of DeepLIFT on handwritten digit images. The darker the area is, the more important that area is. For digit zero, the top part is important since this indicates that

this is a closed circle. For digit three, the middle part is the key to differentiate three from eight. However, this kind of explanation is not completely in correspondence with human intuition, and it is quit subjective to tell whether this is an important part to identify this digit. For example, it is confusing that instead of top and bottom part, why the middle part of digit one is of great importance. It is also unclear that whether the northeastern corner of digit seven is a crucial part of identifying seven.

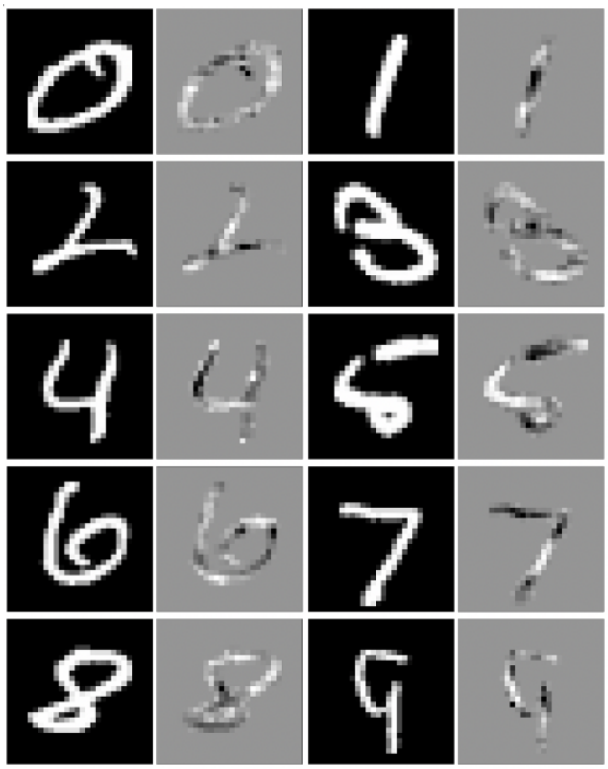

Figure 9: Explanation of DeepLIFT on Handwritten Digit Images

Figure 10 shows the explanation of DeepLIFT on fashion product images. We can see that this explanation plots the outline of that product. It is hard to figure out the important part from this explanation.

4.3 SHAP

I select three explainers from SHAP, DeepExplainer, GradientExplainer, and KernelExplainer. How these three explainers perform on handwritten digit images?

Figure 11 illustrates explanation of DeepExplainer on handwritten digit images. The red area means that this area will increase the probability of this image being in this class, while the blue area indicates that this area will decrease the probability of this image being in this class. For example, when DeepExplainer tries to explain why digit four image is in digit four class, the red area on top indicates that this area makes this image more "four". But when it tries to explain why this image is not in digit nine class, this area turns to blue, which means this area makes this image less "nine". It is significantly reasonable since this is a key part that differentiate digit four from digit nine. We can also see this pattern for digit nine image.

Figure 12 shows explanation of GradientExplainer on handwritten digit images. We can notice that

Figure 10: Explanation of DeepLIFT on Fashion Product Images

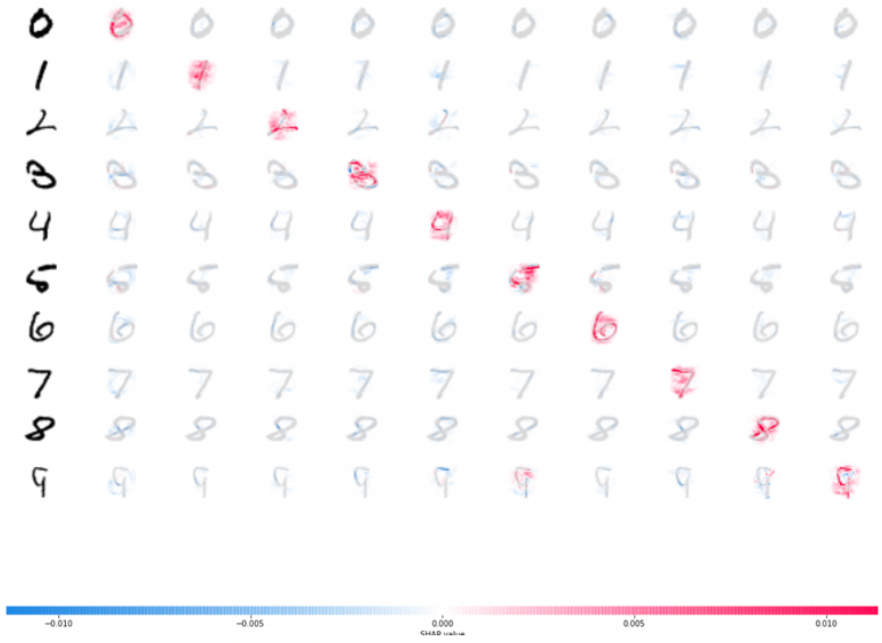

Figure 11: Explanation of DeepExplainer on Handwritten Digit Images

this explanation is similar to that of DeepExplainer, probably because they are both gradient based algorithms.

Figure 13 are explanation of KernelExplainer on handwritten digit images. It is difficult to understand this explanation, and the segmentation is unreasonable.

Then how these three explainers perform on fashion product dataset?

Figure 14 shows how DeepExplainer explains fashion product images. The first column shows the original images, and the rest explain why this image is in this class or not with order backpacks, belts, briefs, casual shoes, and flip flops. The first three images belong to flip flops class, and the

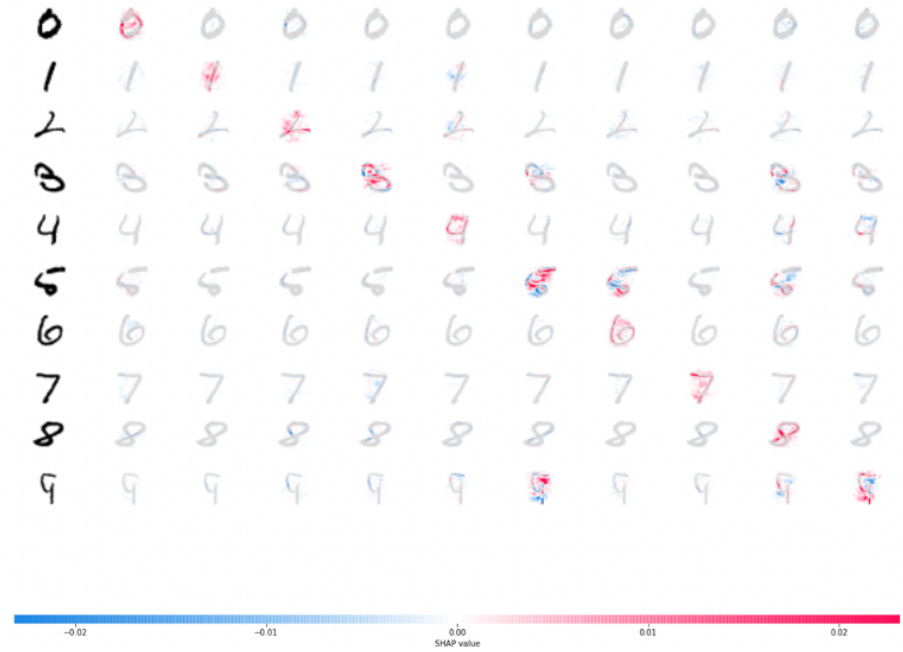

Figure 12: Explanation of GradientExplainer on Handwritten Digit Images

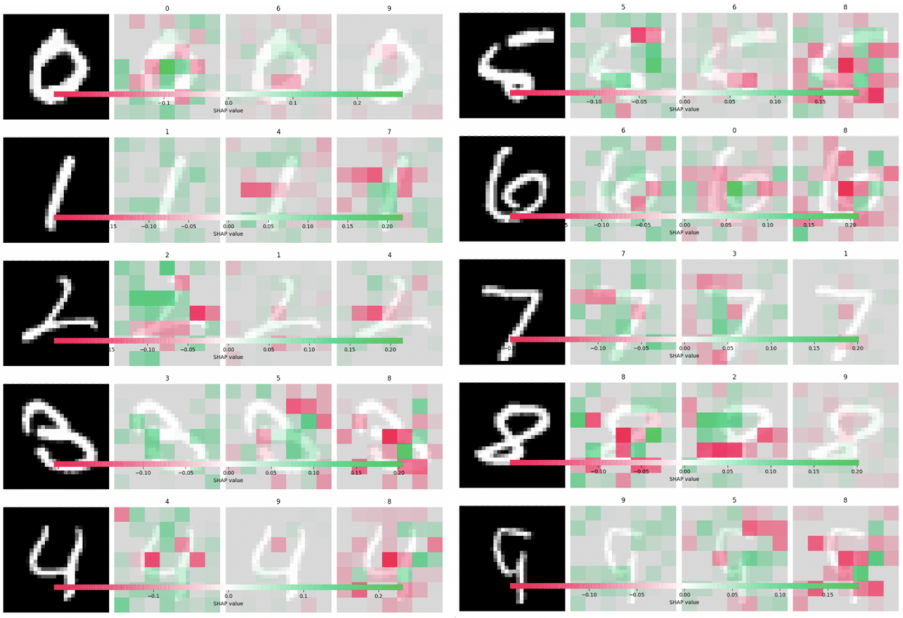

Figure 13: Explanation of KernelExplainer on Handwritten Digit Images

forth column explains why this image is not casual shoes. We can notice that the blue area is located at the end of these shoes, and this reveals that missing of this part decreases the probability of this image being casual shoes, or we can say, makes it less "casual shoes". On the other hand, the fifth column shows why this image is in flip flops class. Similarly, we can see why the last three images are casual shoes instead of flip flops. It is interesting to notice that the model captures the top line of casual shoes including shoelaces as important features. When it tries to explain why this is not flip flops, it focuses on the left side of shoes. Since shoelaces and heels are key features to differentiate casual shoes and flip flops, DeepExplainer can explain a model to a degree that is in correspondence with human tuition.

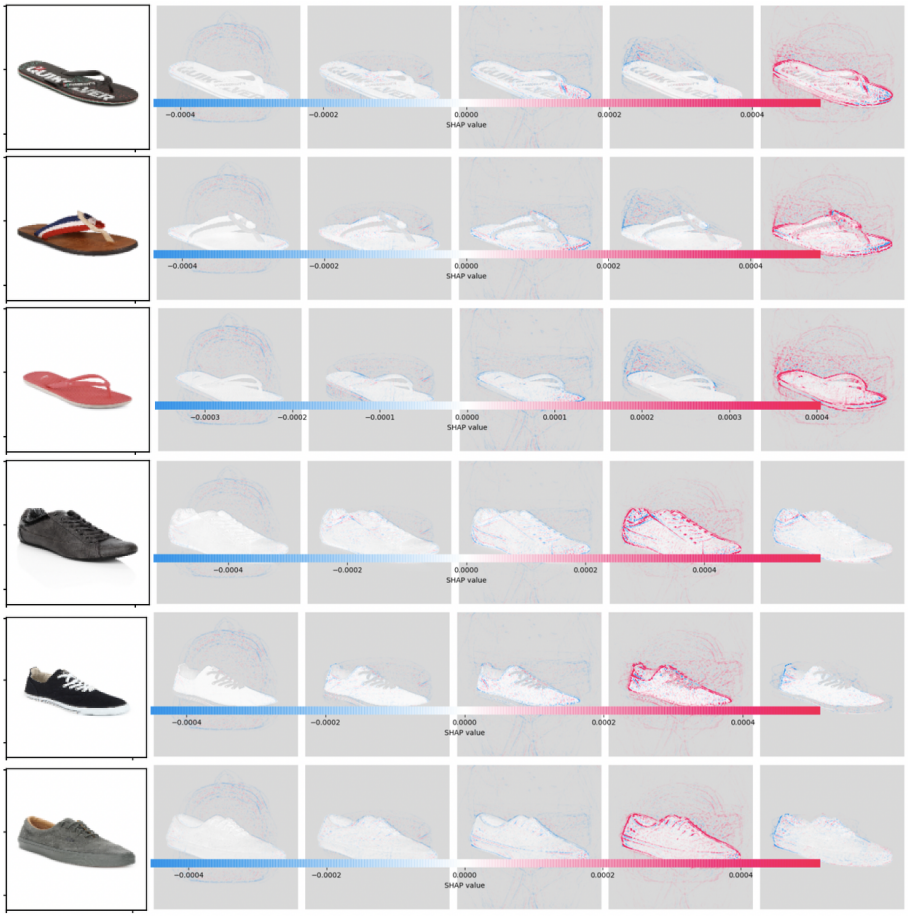

Figure 14: Explanation of DeepExplainer on Fashion Product Images

We can conclude from Figure 15 that GradientExplainer does not perform as good as DeepExplainer on fashion product images, because even though it can capture some key parts, e.g., heels, the blue and red areas are sparse and not as clear as DeepExplainer.

We can see from Figure 16 that KernelExplainer would not be a reasonable explainer on fashion product images, since it credits to background as important features.

4.4 LIME

Figure 17 shows explanation of LIME on handwritten digit images, and Figure 18 shows its explanation on fashion product images. There are three segmentation methods, felzenszwalb, quickshift, and slic, and they are utilized at the first step of LIME when it segments one image into superpixels. After trying these three segmentation methods, I find that slic has the best performance. In addition, from Figure 18, we can see that LIME can identify some important parts including shoelaces and heels, but its explanation of flip flops can be obscure. This may indicates some degree of instability in LIME, and this can be further proved in the next section.

Figure 15: Explanation of GradientExplainer on Fashion Product Images

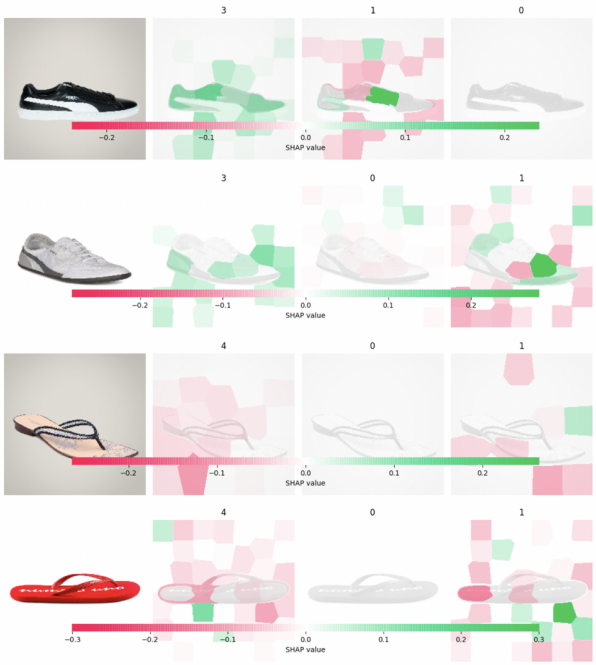

Figure 16: Explanation of KernelExplainer on Fashion Product Images

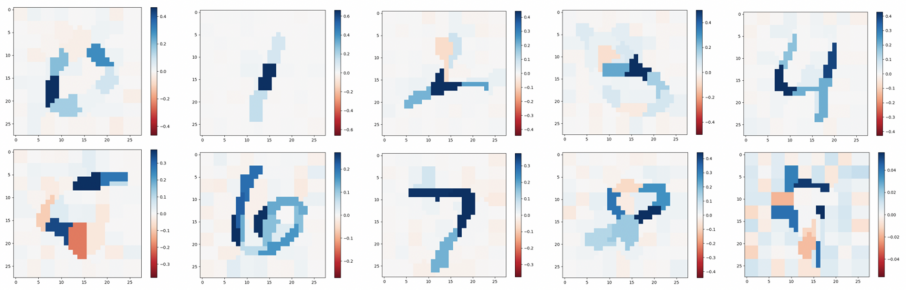

Figure 17: Explanation of LIME on Handwritten Digit Images

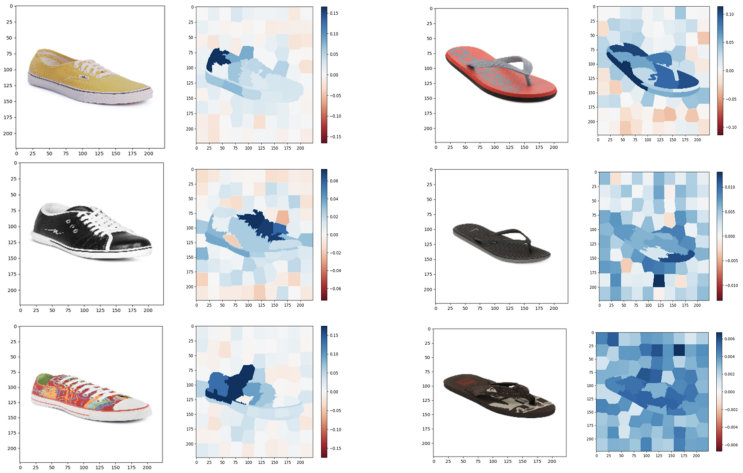

Figure 18: Explanation of LIME on Fashion Product Images

5 Comparison

In order to perform a clear and straightforward comparison, I randomly select two digit images, four casual shoes images, and four flip flops images, and compare their explanation using these three techniques.

As shown in Figure 19, interpretation of digit seven can be similar and meanwhile be slightly different. For example, Deeplift and LIME think the northeastern corner of this seven digit is an important part, while DeepExplainer and GradientExplainer do not hold the same opinion. Besides, DeepLIFT puts emphasis on the bottom part of digit eight, while the others do not think so. On the other hand, all of them think the middle part of digit eight is crucial, and that the beginning and ending parts of digit seven are important. However, since it is also hard to tell whether this part is important and there is no metrics designed to measure goodness of explanation methods, I could not tell which method is preferable over another.

Overall I think DeepExplainer has the best performance since its explanation is in correspondence with human tuition. From Figure 20 and Figure 21, we can illustrate that DeepExplainer can spot shoelaces and heels that are key differences between casual shoes and flip flops. Additionally, we can notice some problem in the other explanation methods. There may be a problem of sparseness in GradientExplainer and instability in LIME, and KernelExplainer sometimes looks at background.

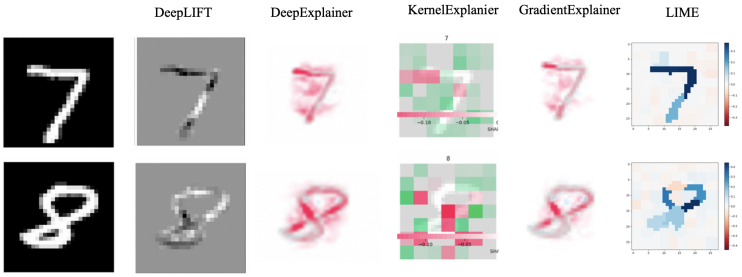

Figure 19: Comparison Based on Handwritten Digit Images

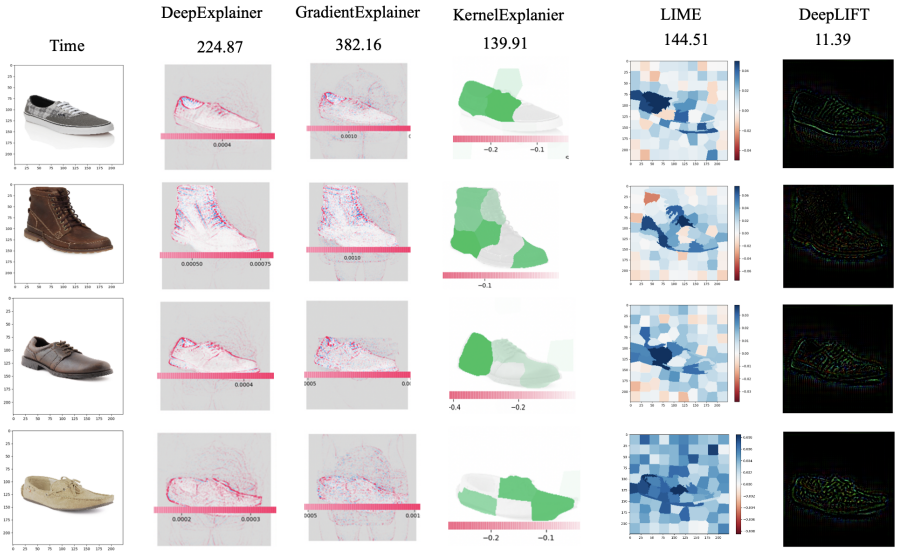

Figure 20: Comparison Based on Casual Shoes

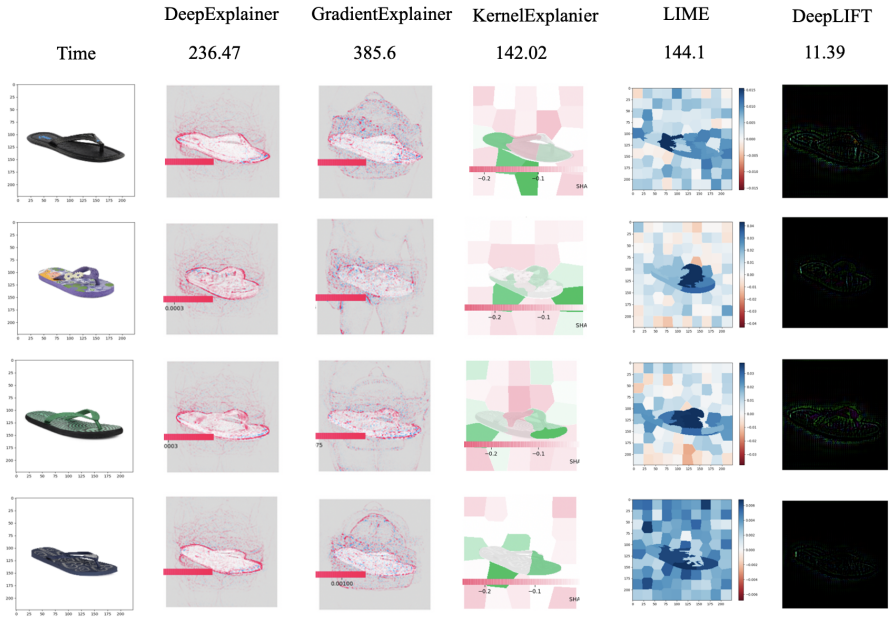

Figure 21: Comparison Based on Flip Flops

6 Conclusion

This project aims to compare the performance of three local explanation methods, DeepLIFT, SHAP, and LIME on two CNN models, thus to conclude which method is preferable over another. First I train two CNN models on handwirtten digit dataset and fashion product dataset. Then after showing the explanation of these three methods on two models, I further compare them based on same images. Even though they can all capture some important features, there are some differences and problems among them. There may be a problem of sparseness in GradientExplainer, and an issue of instability in LIME. KernelExplainer tends to recognize background as important parts. Overall, I think explanation of DeepExplainer correspond with human tuition.

Acknowledgments

I would like to thank my professor, Amir Jafari, for helpful guidance and feedback. I could not have done this work without his help.

7 References

[1] Shrikumar, Avanti and Greenside, Peyton and Kundaje, Anshul, Learning important features through propagating activation differences, arXiv preprint arXiv:1704.02685, 2017

[2] Mohammadreza Salehi, A Review of Different Interpretation Methods in Deep Learning (Part 2: Pixel-wise Decomposition, DeepLIFT, LIME)

[3] Tulio Ribeiro, Marco and Singh, Sameer and Guestrin, Carlos, " Why Should I Trust You?": Explaining the Predictions of Any Classifier, arXiv, arXiv–1602, 2016

[4] Lundberg, Scott M and Lee, Su-In, Advances in Neural Information Processing Systems, 4765–4774, Curran Associates, Inc., 2017